

USB flash memória HOGYAN

Niko Sauer
Unit of Advanced Study
University of Pretoria, South Africa

nikos@friedrichs.up.ac.za

Ezen dokumentum eljárásokat mutat be USB flash memória eszközök (memory stick; memóriakártya) telepítésére, valamint arra, hogyan formázzuk őket különböző fájlrendszerekre, mint például vfat (ahogy általában kapjuk őket) és ext2 (ami jobban illeszkedik a Linuxon való használathoz). Továbbá bemutatjuk hogyan particionáljuk az egységet két részre különböző fájlrendszerekkel.

Tartalomjegyzék

1. Cél.....	3
2. Fizikai eszközök	3
3. Memóriakártya vásárlása	4
4. Ajánlott olvasmányok	4
5. Hardver és rendszermag	5
5.1. A rendszer	5
5.2. Rendszermag beállítások.....	5
5.3. Jegyzetek	6
6. Néhány alapgfogalom	6
6.1. A /proc fájlrendszer	6
6.2. SCSI eszközök.....	6
6.3. Az USB eszköz fájlrendszer.....	7
6.4. Az ext2 fájlrendszer.....	7
6.5. A vfat fájlrendszer	7
6.6. Parancs opciók magyarázata	7
7. Alapvető ellenőrzések	8
7.1. Gyors ellenőrzés	8
7.2. A /proc fájlrendszer tesztelése.....	8
7.3. Az USB fájlrendszer felcsatolása	8
7.4. Vfat tesztek.....	9

8. Ext2 fájlrendszer létrehozása.....	9
8.1. Particionálás	10
8.2. Az eszköz formázása.....	11
8.3. Teszt - ext2	11
9. Két partíció létrehozása.....	11
9.1. Particionálás	12
9.2. Az ext2/vfat rendszer létrehozása.....	13
9.3. Teszt - ext2 és vfat.....	13
9.4. A Linux beállítása a kettős partícióhoz	13
10. A memóriakártya hozzáférhetővé tétele	14
11. Karbantartás	15
11.1. A fájlrendszer vizsgálata	16
11.2. És eztán?.....	17
12. A 2.6-os Linux rendszermag	17
12.1. Mik az újdonságok?	17
12.2. Rendszermag beállítások.....	17
12.3. Megjegyzések	18
12.4. Gyorscsere és elnevezés	18
13. Formaságok	20
13.1. Szerzői jog.....	20
13.2. A felelősség teljes elhárítása	20
13.3. Készítők.....	21
13.4. Visszajelzés	21
13.5. Magyar fordítás	21

1. Cél

A dokumentum célja, hogy leírja az USB flash memória eszközök (memory stick; memóriakártyák) Linuxon való használatát. Ez az alábbiakból áll:

- új eszköz vásárlása, előre formázva MS Windows rendszerekhez, működik Linuxon;
- az eszköz ext2 fájlrendszerre formázása és beüzemelése. Azokban az esetekben, melyekben az eszközt kizárólag Linuxos gépen tervezzük használni.
- két partíció ext2 és vfat fájlrendszerre formázása és beüzemelése. Ha Linux-Linux és Linux-Windows közötti adatcsere is lehetséges.
- az eszköz felhasználó baráttá tétele.
- az eszköz fájlrendszerének karbantartása.
- Linux-2.6.x újabb funkcióinak használata.

A dokumentumban bemutatott eljárások működnek bármely olyan USB memória eszközön, ami képes írható-olvasható egységként üzemelni. A csak olvasható eszközök nem tartoznak ebbe a kategóriába. Az eljárások nem fognak működni, ha az eszköz írási képessége ki van kapcsolva (azaz, ha a írásvédő kapcsoló aktív állásban van).

A formázásra és karbantartásra bemutatott eljárások (elvileg) alkalmazhatók más fájlrendszereken is, például ext3 és reiserfs. Ezeket azonban nem próbáltuk ki, és nem is teszteltük. Nem szenteltünk figyelmet a módszerek más operációs rendszereken történő alkalmazására sem.

Figyelem

A digitális kamerákhoz használt flash memória újraformázása veszélyes lehet. Előfordulhat, hogy a kamera nem fogja felismerni. Használja a kamera beépített menüjét a művelethez.

2. Fizikai eszközök

Flash memória eszközök főleg kis méretű, hordozható egységekként vagy digitális kamerák részeként érkeznek. Gyakori elnevezéseik *memory stick* (memóriakártya) vagy *kulcstartó*. Ezek tartalmukat *nem veszítik el* kikapcsoláskor (non-volatile devices), működési elvüket tekintve elektromos töltéseket használnak az adatok bináris formában való tárolásához. A töltések szinte akármeddig változatlanul maradhatnak, de a változtatások (például írás) korlátozzák az eszköz élettartamát (100000 írás / 8MB).

A memóriakártyák közvetlenül az USB-porthoz csatlakoznak a számítógép hátoldalán (esetleg előlapján vagy oldalán - a lektor). Energiaigényüket az USB-port látja el. Néha kényelmes az USB hosszabbító kábel használata, hogy könnyen elérhető közelségbe hozzuk az eszközt.

A memóriakártyák általában rendelkeznek *írásvédő kapcsolóval*, amit ajánlatos kikapcsolni, hogyha írható-olvasható eszközként akarjuk használni. Ha a kapcsoló "be" állásban van, az eszköz *csak olvasható*. Az eszközön általában van egy LED (Light Emitting Diode; fénykibocsátó dióda), ami jelzi a működőképés állapotot. A LED villogása az adatforgalmat mutatja.

A digitális kamerák memóriakártyát használnak a képek tárolására. Ezeket a kamera látja el árammal, és egy csatlakozókábel segítségével lehet a számítógép USB-portjához kötni. A számítógéppel való összekötéshez a kamerát be kell kapcsolni az átvitel idejére, és egyszerűen ki kell kapcsolni miután az eszközt leválasztottuk. Fontos

megjegyezni, hogy a memóriakártya a kamera akkumulátorából kapja az áramot, ezért a műveletet minél hamarabb fejezzük be.

3. Memóriakártya vásárlása

Fontos szempontok memóriakártya vásárlásához:

A memória mérete

Ez attól függ mire akarjuk használni. 128MB népszerű választásnak tűnik.

Működik Linux alatt?

Olvasd az elismert márkák weboldalán... Vagy kérdezzük meg a forgalmazót. Legyünk óvatosak: lehet, hogy az eladó nem fogja érteni miről beszélünk. A használati útmutató szintén említhet valami ilyesmit: működik Linux-2.4 alatt, vagy lehet benne kép a Linux pingvinről. Ha nem járunk sikerrel ennek kiderítésében, okosabb ha nem vesszük meg.

Hosszabbító kábel és kulcstartó

A memóriakártyákat általában hosszabbító kábellel (2) és (nyakba akasztható) kulcstartóval csomagolják. Az utóbbi biztonságos módját nyújtja az eszköz szállításának. Bizonyosodjunk meg róla, hogy a fentiek megvannak. A körülményektől függően szükség lehet még egy hosszabbító kábelre. Laptop és notebook számítógépekhez általában nem szükséges.

Fájlrendszer

A gyári új memóriakártyák általában vfat (msdos) fájlrendszerre vannak formázva, és működnek Linuxon is, de így nem lehet kihasználni az ext2 (vagy más) fájlrendszer stabilitását és sokoldalúságát. Ez nem probléma, helyettesíthető más fájlrendszerekkel, mint később látni fogjuk.

Ár

A memóriakártyák ára egy adott méreten belül is jelentősen különbözhet. Jó ötlet utánanézni mekkora a szórás, erre kiválóan megfelel az internet. (Az ár nagyban függ a nyújtott szolgáltatásoktól is - a lektor)

4. Ajánlott olvasmányok

Ezen dokumentum olvasóinak ajánlatos megismerkedni az alábbiakkal: ¹

Az alábbi eszközök kézikönyv oldalai: mount, umount, fdisk, mke2fs, mkdosfs, dumpe2fs és fsck.ext2.

Linux documentation project (<http://www.tldp.org>), (Magyar LDP (<http://tldp.fsf.hu>))

USB-Digital Camera HOWTO (<http://www.tldp.org/HOWTO/USB-Digital-Camera-HOWTO>)*, (USB digitális fényképező HOGYAN (<http://tldp.fsf.hu/HOWTO/USB-Digital-Camera-HOWTO-hu/index.html>))

The Linux Partition HOWTO (<http://www.tldp.org/HOWTO/Partition>)* (Linux particionálás HOGYAN)

The Linux 2.4 SCSI subsystem HOWTO (<http://www.tldp.org/HOWTO/The Linux SCSI-2.4-HOWTO>) (Linux 2.4 SCSI alrendszer HOGYAN)

The Linux Kernel HOWTO (<http://www.tldp.org/HOWTO/Kernel-HOWTO>), (Linux-rendszer HOGYAN (<http://tldp.fsf.hu/HOWTO/Kernel-HOWTO-hu/index.html>))

The Linux USB subsystem (<http://www.linux-usb.org/USB-guide/book1.html>)* (Linux USB alrendszer)
Linux filesystem hierarchy (<http://www.tldp.org/guides.html>) (Linux fájlrendszer hierarchia)
The Linux Kernel (<http://www.tldp.org/guides.html>) (A Linux rendszermag)
Linux Kernel 2.4.xx/Documentation (megtalálható az `/usr/src/linux/Documentation` könyvtárban)

5. Hardver és rendszermag

5.1. A rendszer

Itt olvashatók a konfiguráció főbb jellemzői, melyen az alább részletezendő módszerek kifejlesztésre és kipróbálásra kerültek (Linux-2.6 rendszermagon is). A képernyő-ábrák pontos másolatai a monitoron megjelent tartalomnak.

Hardver: Intel (R) Celeron (TM) 1100 MHz
Disztribúció: RedHat Linux 7.0 (alaposan módosítva)
Rendszermag: Linux-2.4.20 (www.kernel.org). Lásd még 12 Linux-2.6.x-hoz.
Segédprogramok: util-linux-2.11z (mount, umount, fdisk); e2fsprogs-1.32 (mke2fs, dumpe2fs, fsck.ext2);
mkdosfs-2.2

5.2. Rendszermag beállítások

Bizonytalan, hogy a 2.4.xx-nél korábbi rendszermagok (kernel) USB támogatása megfelelő-e, ezért külön belefodítottuk a rendszermagba ezt a - dokumentum szempontjából fontos - támogatást. A moduláris megoldás szintén megfelelő.

5.2.1. SCSI támogatás

SCSI támogatás (CONFIG_SCSI scsi_mod.o)
SCSI lemez támogatás (CONFIG_BLK_DEV_SD sd_mod.o)

5.2.2. Fájlrendszerek

DOS FAT fájlrendszer támogatás (CONFIG_FAT_FS fat.o)
MSDOS fájlrendszer támogatás (CONFIG_MSDOS_FS msdos.o)
VFAT (Windows 95) fájlrendszer támogatás (CONFIG_VFAT_FS vfat.o)
/proc fájlrendszer (CONFIG_PROC_FS)
ext2fs fájlrendszer támogatás (CONFIG_EXT2_FS ext2.o)

5.2.3. USB támogatás

USB támogatás (CONFIG_USB usbcore.o)
Előzetes USB eszköz fájlrendszer (CONFIG_USB_DEVICEFS)
USB háttértár (mass storage) támogatás (CONFIG_USB_STORAGE usb-storage.o)

5.3. Jegyzetek

Az alábbi listában a nagybetűs részek a `.config` fájlban levő változóneveket jelentik. Ezt a fájlt a rendszermag-forrás gyökerkönyvtárában lehet megtalálni (`/usr/src/linux/`). Az `xxx.o` részek a modulra utalnak, melyek a moduláris megközelítés során keletkeznek. Ha nincs hivatkozás egy modulra, akkor az az opció kizárólag rendszermagba fordítva érhető el.

Különböző rendszermag verziók más-más módon jelölhetik az egyes beállításokat, például a `make menuconfig` vagy `make xconfig` használatakor. Ezért a változók, mint például `CONFIG_USB`, megbízhatóbb jelölések lehetnek. Ezeket az adott opció *súgójából* tudhatjuk meg.

A naprakész Linux disztribúciók, mint például *RedHat* és *SuSE*, valószínűleg már tartalmazzák a megfelelő beállításokat belefordítva a rendszermagba.

Az USB támogatás alatt számos digitális fényképezőgép beállításait elérhetjük.

Ajánlatos utánanézni a 4. pontban felsorolt írásokban, ha fontolóra vesszük a rendszermag (újra)fordítását.

6. Néhány alapfogalom

Ebben a fejezetben néhány linuxos alapfogalmat tekintünk át röviden, melyek segíthetik a lentebb részletezett módszerek megértését.

6.1. A /proc fájlrendszer

A /proc fájlrendszer ablakként szolgál, melyen keresztül láthatjuk egy linuxos rendszer működését. A dokumentum szempontjából legfontosabb objektumok a következő könyvtárak: `/proc/bus/usb/` és `/proc/scsi/`. Ezek segítségével fogjuk ellenőrizni, hogy a rendszermag céljainknak megfelelően, helyesen van-e beállítva (7.2).

6.2. SCSI eszközök

A memóriakártya egy USB adattároló eszköznek tekinthető, ami a rendszer felé egy cserélhető SCSI lemez (sd). Az SCSI lemezes eszközök a `/dev` (eszközök) könyvtárban `/dev/sda`, `/dev/sdb`, ... néven érhetőek el. Ha több különböző egységünk van, ezek sorban a `/dev/sda`, `/dev/sdb` stb. eszközökhöz lesznek rendelve. Ha például egy memóriakártya és egy digitális fényképezőgép van csatlakoztatva, akkor az egyik kapja a `/dev/sda`-t, a másik pedig a `/dev/sdb`-t. A tesztek azt mutatják, hogy az első észlelt eszköz lesz `sda` és a `/proc/scsi/usb-storage-0` könyvtár fogja képviselni. Azt viszont nem lehet tudni, hogy mi történik, ha bootolás közben mindkét eszköz jelen van. A `/proc/partitions` fájlban meg lehet nézni a partíciók listáját, benne `sda`, `sda1`, `sdb` bejegyzésekkel. A 2.6-os rendszermag sorozatban ezt a problémát nagyon elegánsan oldják meg (12).

A továbbiakban feltételezzük, hogy csupán egyetlen memóriakártya van, és az a `/dev/sda`-hoz van rendelve.

A legtöbb disztribúcióban ezek a pontok megvannak. Az alábbi paranccsal leellenőrizhető, hogy nálunk is léteznek-e: `ls /dev/sda*`. Ha nincsenek, létrehozhatjuk őket az alábbiakkal (root felhasználóként, ezt jelzi a kettős kereszt).

```
# mknod /dev/sda b 8 0
# mknod /dev/sda1 b 8 1
# mknod /dev/sda2 b 8 2
```

stb, egészen `/dev/sda15`-ig, ha szükséges. Az eszköz egy egészként a `/dev/sda`-hoz van rendelve, és a `/dev/sdax` ($x = 1 \dots 15$) jelentik a különböző partíciókat. Később leírjuk hogyan lehet különböző fájlrendszerekhez különböző partíciókat készíteni (lásd. 8 és 9). Ha a memóriakártyát csak egyetlen partícióval (fájlrendszerrel) akarjuk használni, a `/dev/sda1` elegendő lesz.

6.3. Az USB eszköz fájlrendszer

Ezt a dinamikusan generált fájlrendszert a `/proc/bus/usb/` könyvtárba lehet felcsatolni, ezért elengedhetetlen, hogy ez a könyvtár létezzen. Mikor fel van csatolva, több mindent lehet látni a `/proc/bus/usb/` és a `/proc/scsi/` könyvtárakban (7.3). A `/proc/bus/usb/devices` fájlból megtudhatjuk melyik USB eszköz van felcsatlakoztatva (**less** `/proc/bus/usb/devices`). Kis erőfeszítést igényel, hogy megértsük a képernyőn megjelenő szöveget, de azért nem nehéz. A memóriakártya mint *Mass Storage Device* van jelölve.

6.4. Az ext2 fájlrendszer

Az ext2 fájlrendszer még mindig a legelterjedtebb Linux rendszereken. Igen sokoldalú és kifinomult, felvértezve engedélyekkel (olvasás-írás-végrehajtás, kinek van engedélye, hogy csináljon valamit), tulajdonlással (felhasználó, csoport, mások), időbélyeggel (mikor volt utoljára módosítva) stb. Ezenfelül vannak segédeszközei egy ext2-vel ellátott eszköz komplett karbantartásához (11). Ha egy flash memória eszközt kizárólag linuxos gépeken használunk érdemes azt ext2-re formázni (lásd. 8).

6.5. A vfat fájlrendszer

A vfat engedélyezésével a rendszeremben, lehetővé válik Dos/Windows alatt készített fájlrendszerek felcsatolása. A legtöbb memóriakártya Windowson való használathoz van formázva, ezért tekinthetjük vfat-formázottnak. A vfat fájlrendszer kevésbé kifinomult mint az ext2, ezért sokkal gazdaságosabb a memóriát illetően. A jogosultságok másként vannak, mint ext2-ben, ami azt eredményezi, ha egy ext2 fájlt vfat-re mentünk, az újra megjelenik más engedélyekkel. Ennek ellenére, ha a memóriakártyát linuxos és windowsos gépek közötti adatcserére használjuk, a legjobb amit tehetünk, hogy meghagyjuk vfat formában. A kompromisszumot az jelenti, ha kétfelé particionáljuk a memóriakártyát: egy vfat és egy ext2 fájlrendszerrel. Legalábbis a linuxos gépek képesek kezelni mindkettőt. Bővebben: 9 .

6.6. Parancs opciók magyarázata

A dokumentumban felhasznált parancsok rövid listája:

mount -t ext2 /dev/sda1 /mnt/memstick Felcsatolja az sda1 eszközt ext2 fájlrendszerrel az /mnt/memstick könyvtárba.

ls -l Teljes listázás (módok, tulajdonos stb.)

mkdosfs -F 32 /dev/sda1 FAT32 Ms-dos fájlrendszer készítése a /dev/sda1 partíción.

ln -s /dev/sda1 /dev/flash Szimbolikus hivatkozást készít a létező /dev/sda1 partícióhoz /dev/flash néven.

mkdir -m 777 /mnt/memstick/superdir Létrehoz egy új könyvtárat, melyre mindenki rendelkezik rwx jogokkal.

dumpe2fs -h /dev/sda1 Kírja a /dev/sda1 ext2 partíció fejlécét.

7. Alapvető ellenőrzések

Fontos: Mielőtt hozzákezdnenénk, távolítsuk el a hasonló eszközöket az USB buszról, azért, hogy biztosan azt az eszközt lássuk és írjuk, amelyiket szeretnénk.

7.1. Gyors ellenőrzés

Lehetséges, hogy a rendszer már fel van készítve USB flash memória eszközök kezelésére. Ennek előtéséhez egyszerűen adjuk ki a `$ mount` parancsot egy x-terminálról. Ha valami hasonló jelenik meg a kimenetben:

```
none on /proc/bus/usb type usbfs (rw)
```

akkor a 7.4 pontnál lehet folytatni, azonban érdemes lehet átnézni a közbenső részt is. Akkor sincs baj, ha a teszt sikertelen.

7.2. A /proc fájlrendszer tesztelése

Néhány dolgot le lehet ellenőrizni a `/proc` könyvtárban, hogy meggyőződjünk arról, a rendszermag tartalmazza a beállításokat, vagy a megfelelő modulok betöltődtek. Először nézzük meg, hogy a `/proc/bus/usb` könyvtár létezik-e. Ha igen, a rendszermagban van USB alrendszer támogatás. Ha nem, akkor a rendszermagot újra kell fordítani megfelelő USB támogatással (lásd. 5.2.3), vagy a frissíteni kell azt. Másodszor a `/proc/scsi` könyvtár meglétét kell ellenőrizni. Ha ott van, minden rendben, ellenkező esetben a SCSI támogatást bele kell fordítani a rendszermagba (lásd. 5.2.1).

A `/proc`-nak rendelkeznie kell az USB fájlrendszer csatolási pontjával. Ez a pont a `/proc/bus/usb`. Ha megvan, a rendszermag helyesen van beállítva.

7.3. Az USB fájlrendszer felcsatolása

Ha az előző rész ellenőrzései sikeresek voltak, következő lépésként csatolni kell az USB fájlrendszert. Ezt root felhasználóként kell megtenni a következőképpen:

```
# mount -t usbfs none /proc/bus/usb
```

Megjegyzés: Régebbi rendszermag-verziókban a fenti mount parancs `usbfs` paraméterét ki kell cserélni `usbdevfs`-re. 2.4.20-as Linuxokban mindkét változat működik.

Ha minden rendben, végezzünk további teszteket. Első a gyors ellenőrzés (7.1). Az alapos ellenőrzéshez adjuk ki a `# ls -l /proc/bus/usb` parancsot, ami valami hasonló kimenetet fog adni:

```
dr-xr-xr-x 1 root root 0 Sep 19 14:21 001
dr-xr-xr-x 1 root root 0 Sep 19 14:21 002
-r--r--r-- 1 root root 0 Sep 19 22:30 devices
-r--r--r-- 1 root root 0 Sep 19 22:30 drivers
```


A `/proc/scsi/usb-storage-0/` könyvtárnak most már léteznie kell, és az egyik benne levő fájl az alábbiakat tartalmazza. Az én rendszeremben a `$ less /proc/scsi/usb-storage-0/1` parancs a következőt írja ki:

```
Host scsil: usb-storage
  Vendor: Generic
  Product: Mass Storage Device
Serial Number: None
  Protocol: Transparent SCSI
  Transport: Bulk
  GUID: 0ed16680000000000000000000000000
  Attached: Yes/No
```

Ha a flash meghajtó csatlakoztatva van, az utolsó sorban a "Yes" bejegyzés olvasható, egyébként pedig a "No".

7.4. Vfat tesztek

Most már készen állunk arra, hogy kiderítsük működik-e a memóriakártya. Feltételezzük, hogy az eszköz gyári új állapotban van. A használati útmutató valószínűleg tartalmazza, hogyan állítsuk be Windowshoz. Van egy jel, hogy vfat fájlrendszerre van formázva. Mielőtt megpróbálnánk felcsatolni, készítsünk egy csatolási pontot neki a következő paranccsal: `# mkdir -m 777 /mnt/memstick` A felcsatolás parancsa:

```
# mount -t vfat /dev/sda1 /mnt/memstick
```

Ha minden simán ment, már láthatjuk az eszközt: `# ls /mnt/memstick`.

Most próbáljunk ki néhány alapvető dolgot, mint például könyvtár létrehozása az eszközön, és egy szöveges fájl másolása:

```
# mkdir /mnt/memstick/apollo
# cp /home/myname/myfavourite_file /mnt/memstick/apollo/.
```

Listázzunk újra (`# ls -l /mnt/memstick`) és figyeljük meg az engedélyeket.

Csatoljuk le az eszközt (`# umount /dev/sda1`) és csatoljuk fel újra ahogy előzőleg is tettük. Listázzunk megint és nézzük meg az engedélyeket. Valószínűleg a szöveges fájlunk van x-jogosultsága, vagyis futtathatóvá vált. Ez teljesen szokványos a vfat fájlrendszerben. Ha elégedett vagy ezzel, csatold le az eszközt, és menj a 10 pontra.

8. Ext2 fájlrendszer létrehozása

Az itt leírt eljárások során az egész flash memória eszközön egyetlen ext2-es partíciót alakítunk ki. Ez alkalmasabbá teszi az eszközt linuxos gépek közötti használatra. Azonban semmiképpen se tegyük, ha Windowson is használni akarjuk.

Fontos: A `/dev/sda` könyvtárba felcsatolt eszköz újra lesz formázva és az összes adat törlődik róla. Mielőtt hozzákezdene, távolítsuk el a hasonló eszközöket az USB buszról, azért, hogy biztosan azzal az eszközzel dolgozzunk, amellyel szeretnénk.

8.1. Particionálás

Feltételezzük, hogy a flash memória a `/dev/sda` könyvtárba van felcsatolva. Ebben a fejezetben azt az esetet nézzük, melyben az egész eszköznek egyetlen ext2 partíciót készítünk. Az alábbi példában egy 128MB-os flash memóriát formáztunk ext2 fájlrendszerre. Emiatt látható 131MB és 888 cilinder az ábrán. Az fdisk segédprogramot használtuk, ami egyszerűen kiolvassa ezeket az adatokat az eszközből.

Az összes műveletet root felhasználóként végezzük. A partíciót a `/dev/sda` egységen alakítjuk ki. (nem a `/dev/sda1-en`). Az eljárást magyarázatokkal tűzdelt lépések sorozataként mutatjuk be. Az fdisk alap promptja `Command (m for help):`, és bármely fázisnál üthetünk `m`-et az érvényes parancsok megtekintéséhez. Ha ezt tesszük, az eredmény a következő lesz:

```
Command  action
a        toggle a bootable flag
b        edit bsd disklabel
c        toggle the dos compatibility flag
d        delete a partition
l        list known partition types
m        print this menu
n        add a new partition
o        create a new empty DOS partition table
p        print the partition table
q        quit without saving changes
s        create a new empty Sun disklabel
t        change a partition's system id
u        change display/entry units
v        verify the partition table
w        write table to disk and exit
x        extra functionality (experts only)
```

A memóriakártyát csatlakoztassuk, de ne csatoljuk fel. Ügyeljünk arra, hogy az írásvédő ki legyen kapcsolva.

Az eljárás:

```
# fdisk /dev/sda
Command (m for help):d {enter}
Selected partition 1
Command (m for help):n {enter}
Command action
  e  extended
  p  primary partition (1-4)
p {enter}
Partition number (1-4):1 {enter}
First cylinder (1-888, default 1): {üss enter-t}
Using default value 1
Last cylinder ... (1-888, default 888): {üss enter-t}
```

```
Using default value 888
```

Most ellenőrizhetjük a partíciós tábla kiíratásával, hogy minden rendben ment-e.

```
Command (m for help): p
Disk /dev/sda: 131 MB, 131072000 bytes
9 heads, 32 sectors/track, 888 cylinders
Units = cylinders of 288 * 512 = 147456 bytes

Device      Boot Start End Blocks Id System
/dev/sda1           1   888 127856 83 Linux
```

Győződjünk meg róla, hogy a partíciót nem állítottuk be bootolhatónak, ekkor nincs csillag a partíciós tábla "Boot" oszlopában. Továbbá az "Id" és a "System" megegyeznek a fenti táblázatban levőkkel. Ez azt mutatja, hogy az eszközt ext2-re formázhatjuk (következő rész). Ezek az alapértelmezett értékek. Ha valami nem stimmel, akkor megváltoztathatjuk a következő parancsokkal:

```
Command (m for help): a [toggle a bootable flag]
Command (m for help): t [change a partition's system id]
```

Ha (vagy amikor) a partíciós tábla helyes, az eljárás befejezéséhez:

```
Command (m for help): w [write table to disk and exit]
```

Meg is van!

8.2. Az eszköz formázása

A particionálás befejezése után egyenesen az eszköz formázásához érkeztünk. Ehhez az mke2fs segédprogramot fogjuk használni:

```
# mke2fs /dev/sda1
```

A memóriakártya LED-je ezalatt folyamatosan villog. Mikor abbahagyja, a program végzett.

8.3. Teszt - ext2

A formázás sikerének ellenőrzéséhez ismételjük meg a 7.3 és a 7.4 fejezetben leírtakat két kis eltéréssel. Az első eltérés az, hogy a mount parancs a következő:

```
# mount -t ext2 /dev/sda1 /mnt/memstick
```

A második különbség, hogy a szöveges fájl engedélyei ezentúl nem módosulnak.

9. Két partíció létrehozása

9.1. Particionálás

A bemutatásra kerülő eljárás során a flash memória eszközt kétfelé particionáljuk, az egyiket ext2-re, a másikat pedig vfat-re formázzuk. Ez kényelmes lehet azokban az esetekben, mikor két (vagy több) linuxos (ext2) gép, valamint linuxos és windowsos gépek (vfat) között cserélünk adatokat.

Fontos: A `/dev/sda` könyvtárba felcsatolt eszköz újra lesz formázva és az összes adat törlődik róla. Mielőtt hozzákezdenénk, távolítsuk el a hasonló eszközöket az USB buszról, azért, hogy biztosan azzal az eszközzel dolgozzunk, amellyel szeretnénk.

Ebben a részben bemutatjuk, hogyan hozunk létre két partíciót a memóriakártyán: az egyik felét vfat-nek, a másikat pedig az ext2-nek. Így sokoldalúbb lesz az eszköz, de kicsit megosztott is. Az eljárás nem lesz olyan részletes, mint a 8.1 fejezetben. Az ott leírtak lesznek ismételve, valamint mindkét fájlrendszer hexadecimális kódja közvetlenül lesz beírva ("t" bejegyzést követő sorok: Win95 = b; Linux = 83). Ha a partíció csak Linuxnak lesz, - szükségtelen mondani - a Linux az alapértelmezés. Továbbá jegyezzük meg, hogy a vfat partíciót hozzuk létre elsőként - a Windows miatt kell így tennünk.

```
# fdisk /dev/sda

Command (m for help): d
Selected partition 1

Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
P
Partition number (1-4): 1
First cylinder (1-888, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-888,
    default 888):444
Command (m for help): t
Selected partition 1
Hex code (type L to list codes): b

Changed system type of partition 1 to b (Win95 FAT32)

Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
P
Partition number (1-4): 2
First cylinder (445-888, default 445):
Using default value 445

Last cylinder or +size or +sizeM or +sizeK (445-888,
```

```

default 888):

Using default value 888

Command (m for help): t
Partition number (1-4): 2
Hex code (type L to list codes): 83

Command (m for help): p

Disk /dev/sda: 131 MB, 131072000 bytes
9 heads, 32 sectors/track, 888 cylinders
Units = cylinders of 288 * 512 = 147456 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1            1           444       63920    b   Win95 FAT32
/dev/sda2            445          888       63936    83   Linux

Command (m for help):w

```

Megjegyzés: Figyeljük meg, hogy a tábla írása előtt a második partíciót is létrehoztuk.

A döntés, hogy pontosan a közepén választottuk ketté az egységet (128MB-os memória kártya 1--444; 445--888) teljesen önkényes volt. Bármely más elgondolás ugyanolyan jó.

A "delete" részt annyiszor kell ismételni, ahány partíció van az egységen.

9.2. Az ext2/vfat rendszer létrehozása

Ez a lépés a 8.2 fejezet ismétlése. Az első lépés létrehozza a DOS-os vfat, a második pedig a linuxos ext2 partíciót.

```

# mkdosfs -F 32 /dev/sda1
# mke2fs /dev/sda2

```

9.3. Teszt - ext2 és vfat

Egyszerűen végezzük el a 7.4 és a 8.3 fejezetben leírt tesztekét külön az egyes partíciókra. A csatolási parancsok:

```

# mount -t vfat /dev/sda1 /mnt/fatstick
# mount -t ext2 /dev/sda2 /mnt/memstick

```

Ezzel felcsatoltuk az eszköz mindkét partícióját.

9.4. A Linux beállítása a kettős partícióhoz

Az a linuxos gép amin a particionálást végeztük, helyesen fogja kezelni a kettős partíciót. Mikor elvisszük az eszközt egy másik linuxos géphez, az csak a `/dev/sda1` partíciót fogja látni. Ez talán egy hibája a rendszernek. A új gépen a következőt kell tennünk:

```
# fdisk /dev/sda
    Choose p; check partition table
    If satisfied, choose w
# less /proc/partitions
```

Az eszköznek így már megvan minden partíciója és rajtuk az adatok érintetlenül. Egyetlen partíció esetén a fentieket nem szükséges elvégezni.

10. A memóriakártya hozzáférhetővé tétele

Eddig csupán a rendszer beállításának és a memóriakártyák formázásának menetét írtuk le. Az USB eszközök fájlrendszereinek felcsatolása rugalmatlan, a nem root felhasználók számára egyszerűbbé kell tennünk az eszköz felcsatolását. Ehhez csupán néhány dolgot kell megtennünk.

Az alábbi eljárás az összetettebb esetre vonatkozik, melyben a memóriakártyát kétfelé particionáltuk (9). Egyetlen ext2 partíció esetén a dolog még egyszerűbb: a vfat-ra vonatkozó sorokat elhagyjuk és az `sda2`-t kicseréljük `sda1`-re.

Feltételezzük, hogy a `/mnt/memstick` és `/mnt/fatstick` csatolási pontok rendelkezésre állnak. Nem kötelező ezeket a neveket használni, és nem kell az `/mnt` könyvtárba rakni őket.

Az emlékezést megkönnyítendő, készítsük el a következő szimbolikus hivatkozásokat:

```
# ln -s /dev/sda1 /dev/fatflash
# ln -s /dev/sda2 /dev/flash
```

A gördülékenyebb és könnyebb felcsatolásért adjuk a következő sorokat az `/etc/fstab` fájlhoz: (lásd. a megjegyzéseket a 7.3 végén)

```
none          /proc/bus/usb usbfs      defaults      0 0
/dev/flash    /mnt/memstick ext2,vfat     rw,user,noauto 0 0
/dev/fatflash /mnt/fatstick vfat         rw,user,noauto 0 0
```

A középső sor engedélyezi a felcsatolást ext2 és vfat esetben is. Az utolsó két sor lehetővé teszi bármely felhasználó számára az eszköz felcsatolását a következő parancsokkal:

```
$ mount /dev/flash [ext2-nek vagy vfat-nek]
$ mount /dev/fatflash [vfat-nek]
```

Lehetséges mindkét partíció egyidejű felcsatolása is. Lásd. 9 fejezet.

Egyetlen ext2 partíció esetén nem tűnik lehetségesnek, hogy bármely (nem root) felhasználó írható-olvasható módban csatolja fel az eszközt. A megoldáshoz készítsünk egy könyvtárat a memóriakártyán, amit a felhasználó birtokol, teljes hozzáféréssel. Root felhasználóként adjuk ki a következő parancsokat:

```
# mount /dev/flash
# mkdir -m 777 /mnt/memstick/superdir
# chown charles:charles /mnt/memstick/superdir
```

Ha az eszköz vfat formátumú, ez a lépés nem szükséges.

Végezetül, íme egy kis szkript, amit a saját (ext2) flash meghajtóm fel- és lecsatolásához használok:

```
#!/bin/bash
EXCODE="keepit"
green='\033[0;32m'
yellow='\033[0;33m'
## _____
## Function to echo in colours
echo_in_color ()
{
    message=$2
    message1=$4
    color=$1
    color1=$3
    echo -e -n $color
echo -n $message
    echo -e -n $color1 " "
    echo -n $message1 " "
    tput sgr0
    return
}
## _____
clear
mount /dev/flash
echo_in_color $green "Flash drive mounted"
sleep 2
while [ $EXCODE != "flexit" ]
do
    clear
    echo_in_color $yellow "Enter [flexit] to unmount and exit:"
    read EXCODE
done
umount /dev/flash
exit
```

Egy FVWM menüből hívom meg az **Exec exec xterm -geometry 43x2+1250+0 -e /home/nikos/bin/flashdrive** paranccsal, ami valamelyik sarokban megjelenő kis x-term ablakban futtatja. A horizontális elhelyezést (1250) a képernyő felbontásához kell beállítani.

11. Karbantartás

11.1. A fájlrendszer vizsgálata

Hasznos szokás a flash memórián levő ext2 fájlrendszer gyakori ellenőrzése. Ehhez használjuk a `dumpe2fs` programot a következő módon: `# dumpe2fs -h /dev/sda1` (root felhasználóként kell felcsatolni az eszközt).

Az eredmény az alábbihoz hasonló lesz:

```

Filesystem volume name: <none>
Last mounted on: <not available>
Filesystem UUID: c42a6963-5e6a-4cd2-b7d7-c8f09dca6c52
Filesystem magic number: 0xEF53
Filesystem revision #: 1 (dynamic)
Filesystem features: dir_index filetype sparse_super
Default mount options: (none)
Filesystem state: clean
Errors behavior: Continue
Filesystem OS type: Linux
Inode count: 32000
Block count: 127856
Reserved block count: 6392
Free blocks: 116456
Free inodes: 31922
First block: 1
Block size: 1024
Fragment size: 1024
Blocks per group: 8192
Fragments per group: 8192
Inodes per group: 2000
Inode blocks per group: 250
Filesystem created: Sat Sep 20 12:43:00 2003
Last mount time: Tue Oct 28 14:13:03 2003
Last write time: Tue Oct 28 14:28:27 2003
Mount count: 13
Maximum mount count: 35
Last checked: Sat Oct 18 11:28:26 2003
Check interval: 15552000 (6 months)
Next check after: Thu Apr 15 11:28:26 2004
Reserved blocks uid: 0 (user root)
Reserved blocks gid: 0 (group root)
First inode: 11
Inode size: 128
Default directory hash: tea
Directory Hash Seed: 118bee0a-efa5-4771-967e-41a0badd0355

```

Néhány fontos szempontot szükségesnek tartunk kiemelni:

- Az ext2 fájlrendszer létrehozásakor megadja, hogy hányszor lehet felcsatolni mielőtt ellenőrizni kellene. Ezt láthatjuk a *Maximum mount count* (35) és a *Check interval* (lejárató idő) sorokban.
- Az eddigi használat: *Mount count* és *Last checked*.

- Hibás fájlok (sérült blokkok) létezése: *Filesystem state*.

A rendszer figyelmeztetéseket küldhet ezekről a dolgokról az eszközről történő olvasáskor vagy felcsatolásakor.

11.2. És eztán?

Amikor a számláló eléri a küszöbértéket, vagy hibás fájlok vannak, futtassuk a `# fsck.ext2 /dev/sda1` programot a felcsatolt eszközön. Ez frissíti a számlálót, valamint kijavítja a hibás blokkokat.

Megjegyzés: Vfat fájlrendszer esetén a dump program nem tűnik jónak. A `# dumpe2fs -f /dev/sda1` parancs nem használható vfat fájlrendszeren. Létezik egy dosfsck nevű program (csupán alfa verzió), de kockázatos nem saját magunk által formázott eszközön futtatni.

12. A 2.6-os Linux rendszermag

12.1. Mik az újdonságok?

A rendszer beállítása a fent leírt módon tökéletes 2.4.x rendszermagok esetén, azonban 2.6.x (x = 0, 1, 2, 3, a készítés időpontjaker) esetén sokkal több lehetőségünk van:

- Az USB eszközök gyorscserejét (hotplugging) nagy mértékben javították, így lehetőségünk van menet közben USB eszközöket fel- és lecsatolni. Ha felcsatolunk, az eszköz megjelenik a /proc fájlrendszerben, az eltávolítás után pedig eltűnik onnan.
- Bemutatkozott a *sysfs* rendszer. Ezt használva a csatlakoztatott eszközök egy szűk köre, azok jellemzői és a rajtuk lévő fájlrendszerek node-jai talán elérhetők.
- A gyorscserehez szükséges részeket tökéletesítették, így már a felcsatolás sorrendjétől függetlenül meg tudjuk különböztetni az eszközöket.

12.2. Rendszermag beállítások

A rendszermag konfigurációs beállításai nagyrészt megegyeznek a 5.2 fejezetben leírtakkal. A beállító menüt azonban (`# make menuconfig`) szisztematikusan átszervezték.

12.2.1. Busz beállítások

Gyorscsérés eszközök támogatása (CONFIG_HOTPLUG)

12.2.2. Eszközmeghajtók

- **SCSI eszköz támogatás**
SCSI eszköz támogatás (CONFIG_SCSI scsi_mod)

hagyományos /proc/scsi támogatás (CONFIG_SCSI_PROC_FS)
SCSI lemez támogatás (CONFIG_BLK_DEV_SD sd_mod)

- **USB támogatás**

USB támogatás (CONFIG_USB usbcore)
USB eszköz fájlrendszer (CONFIG_USB_DEVICEFS)
UHCI HCD támogatás (CONFIG_USB_UHCI_HCD uhci_hcd)
USB Mass Storage támogatás (CONFIG_USB_STORAGE usb_storage)

12.2.3. Fájlrendszerek

ext2 fájlrendszer (CONFIG_EXT2_FS ext2)

- **DOS/FAT/NT fájlrendszerek**

DOS FAT fájlrendszer támogatás (CONFIG_FAT_FS fat)
MSDOS fájlrendszer támogatás (CONFIG_MSDOS_FS msdos)
VFAT (Windows 95) fájlrendszer támogatás (CONFIG_VFAT_FS vfat)

- **Látszólagos fájlrendszerek**

/proc fájlrendszer támogatás (CONFIG_PROC_FS)

12.3. Megjegyzések

A saját érdekedben ajánlatos elolvasni az *Migrating to Linux Kernel 2.6* (<http://thomer.com/linux/migrate-to-2.6.html>) (Áttérés a 2.6-os Linux rendszeremre) leírást, de nem szükséges olyan lépéseket követni, amelyeket nem tudsz.

12.4. Gyorscsere és elnevezés

A 6.2 fejezetben tárgyaltuk az egy időben felcsatolt USB eszközök esetét. A nehézséget az okozza, hogy a felhasználó számára nehéz az eszközök megkülönböztetése. Előrehaladás történt ez ügyben a Linux Hotplug Project (<http://sourceforge.net/search/>)-ben (Linux Gyorscsere Projekt) (keress rá a hotplug-ra), és a 2.6-os rendszermag-családban már meg is valósították, lehetővé téve a probléma hatékony kezelését. Ehhez szükség van néhány dologra:

- sysfs - a Linux-2.6.x-os rendszermagba alapértelmezetten bele van fordítva
- sysfsutils-0.4.0 vagy későbbi verziók (<http://linux-diag.sourceforge.net/Sysfsutils.html>)
- hotplug-2004_01_05 vagy későbbi verziók (<http://sourceforge.net/projects/linux-hotplug>)
- udev-016 vagy későbbi verziók (<http://www.kernel.org/pub/linux/utils/kernel/hotplug>)

12.4.1. Telepítés

A sysfs fájlrendszer.

A `sysfs` fájlrendszer része a 2.6-os rendszermag sorozatnak. Az `udev` és a `sysfstools` használja, hogy információt szerezzenek rendszermag-objektumokról (eszközök stb.): tulajdonságaikat és közös kapcsolatukat. Az aktiválás a következőképpen történik:

```
# cd /
# mkdir sys
# mount -t sysfs none /sys
```

Az `ls -F /sys` parancs kiírja az egész könyvtárstruktúrát bejegyzésekkel, mint például `block/`, `bus/`, ... Ez is sikerült. A felcsatolást tartóssá tehetjük, ha az alábbi sort az `/etc/fstab` fájlhoz adjuk:

```
none /sys sysfs defaults 0 0
```

Sysfsutils.

A szokványos módon telepíthető, nem igényel speciális beállítást. A README fájl útmutatásai jól érthetőek. Azonban elsőként bizonyosodjunk meg arról, hogy az `/usr/local/lib` bejegyzés szerepel az `/etc/ld.so.conf` fájlban. Erre azért van szükség, hogy a `libsysfs.so` programkönyvtárakat felismerje a rendszer. Az telepítés után ellenőrizzük, hogy az `/usr/local/bin/lshbus` és az `/usr/local/bin/systool` segédprogramok települtek-e. Végül pedig ellenőrizzük, hogy a fent említett programkönyvtárak helyesen be vannak linkelve (`ldconfig -p | grep libsysfs`).

Gyorscsere.

Következő lépésként telepítsük a `hotplug` szkripteket. Ha megvan, ellenőrizzük működésüket: `less /proc/sys/kernel/hotplug`. Az eredmény valami hasonló lesz: `/sbin/hotplug`. Ha mégsem, akkor egy megfelelő inicializáló szkriptbe (például: `/etc/rc.d/rc.local`) írjuk be: `echo "/sbin/hotplug" > /proc/sys/kernel/hotplug`. Ezt a parancsot (root felhasználóként) kiadhatjuk magunk is az ellenőrzés előtt. A telepítés különösen egyszerű: `# make install`. Ezután nézzük meg, hogy a `/sbin/hotplug` szkript megvan-e.

Udev.

Ez az utolsó telepítés. Tanulmányozzuk a README fájlt, és kövessük a `make USE_KLIBC=true` futtatásához szükséges lépéseket. Ha már fordítottál 2.6.x-es rendszermagot, nem fog gondot okozni. Ellenőrizzük, hogy az `/sbin/udev` és az `/sbin/udevinfo` léteznek-e. Vannak más fájlok is, az egyik nagyon fontos az `/etc/udev/udev.rules`. Ha van egy működő rendszered, amivel működik a memóriakártyád (ahogy fent bemutattuk), és felcsatolod azt, kipróbálhatod, hogy a rendszer látja-e (`sda`, `sda1`, ...): `ls /udev`. Ha ott van, csatold le és listázz újra. Már nem lesz ott!

12.4.2. Memóriakártyák elnevezése

Az `udev` lehetőséget biztosít a memóriakártyák felismerésére, néhány beépített tulajdonság alapján. Ez a `vendor` azonosító alapján működik, amit a `systool -vb scsi | grep vendor` parancssal tudhatunk meg, ha az eszköz csatlakoztatva van. Egy listát kapunk a csatlakoztatott SCSI eszközökről (emlékezzünk, hogy a memóriakártya a rendszer felé egy SCSI eszköz; 6.2 fejezet). Feltételezzük, hogy egyidejűleg két memóriakártya van felcsatolva. Ekkor két `vendor` azonosítót találunk. Név szerint: "UFD" és "STORAGE". Adjuk a következő sorokat az `/etc/udev/udev.rules` fájlhoz (akár az elejére is lehet):

```
## Flash Memory 1
```

```
BUS="scsi", SYSFS_vendor="UFD*", NAME="namib%n"

## Flash Memory 2
BUS="scsi", SYSFS_vendor="STORAGE*", NAME="kalahari%n"
```

A vendor azonosítóban lehetnek kiegészítő szóközhözök, ezért kellett a csillagok. Ne hagyjuk le őket. Válasszuk el, majd ismét csatlakoztassuk a memóriakártyákat, és adjuk ki a `ls /udev` parancsot. A személytelen `sda`, `sda1` stb. helyett `kalahari`, `kalahari1`, `namib` és `namib1` van kiírva. Ezzel elneveztük az eszközt a "vendor" információ alapján. Mivel most már fel tudjuk ismerni az eszközt, nem probléma többé, hogy mihez van rendelve. Már csak annyi dolgunk van, létrehozuk a szükséges csatolási pontokat, például `/mnt/namib` és `kalahari`, valamint bejegyezzük őket az `/etc/fstab` fájlba:

```
/udev/namib1 /mnt/namib vfat,ext2 user,noauto,rw 0 0
/udev/kalahari1 /mnt/kalahari vfat,ext2 user,noauto,rw 0 0
```

Ez jó móka... És jusson eszünkbe, hogy vannak még más sivatagok is a Földön.

13. Formaságok

13.1. Szerzői jog

This document is copyrighted (c) 2003 Niko Sauer and is distributed under the terms of the Linux Documentation Project (LDP) license, stated below.

Ha nincs másként feltüntetve, a Linux HOGYAN dokumentumok szerzői jogait a szerzőjük jegyzi. A Linux HOGYAN dokumentumok egészben vagy részben másolhatók illetve terjeszthetők, elektronikus vagy más módon, amennyiben ez a szerzői jogi megjegyzés megtalálható minden másolatban. Az üzleti terjesztés is megengedett, azonban a szerző szeretne értesülni minden ilyen kiadványról.

Minden fordításnak, összefoglaló munkának és olyan művek, amely forrásként használ valamely Linux HOGYAN dokumentumot tartalmaznia kell ezt a szerzői jogi megjegyzést: senki sem készíthet HOGYANt felhasználó munkát és arra nem szabhat ki további megszorításokat. Bizonyos feltételek mellett engedélyezhető kivétel ezen szabályok alól; bővebben a Linux HOGYAN koordinátortól az alábbi címen.

Röviden, szeretnénk mindenkit buzdítani az információ elterjesztésére minél több csatornán. Azonban szeretnénk, hogy mindenki megtartsa a copyright megjegyzést a HOGYAN dokumentumokban, továbbá szeretnénk értesülni minden újabb tervezett kiadásról.

Ha kérdésed van, írd a `<nikos@friedrichs.up.ac.za>` címre.

13.2. A felelősség teljes elhárítása

Semmilyen felelősséget nem vállalunk a dokumentum tartalmával kapcsolatban. Mindenki a saját felelősségére használja az elgondolásokat, példákat. Mivel ez egy új dokumentum, lehetnek benne hibák és pontatlanságok, amik akár a rendszer károsodásához is vezethetnek. Ezért mindenki figyelmesen járjon el, és habár ez különösen valószínűtlen, a szerző semmilyen felelősséget nem vállal érte.

Az összes szerzői jog a tulajdonosuk birtokolja, hacsak nincs másként jelezve. Egy szakkifejezés használatát nem tekintjük úgy, hogy hatással lenne bármely márkanév vagy szolgáltatás védjegyének érvényességére.

Bizonyos termékek vagy márkanevek megnevezését nem tekintjük hozzájárulásnak.

Erősen ajánlott biztonsági mentéseket készíteni a rendszerről nagyobb telepítések előtt és rendszeres időközönként.

13.3. Készítők

Szeretném kifejezni mély hálámat fiaimnak Paulnak és Philipnek, akik 1995-ben rábeszéltek a Linuxra való áttérésre. Philip látott el alapos tanácsokkal és nyújtott bepillantást, ami egy kellemes tanulságos tapasztalattá tette ezt a projektet.

Emma Jane Hogbin a tanulmányában felvetett sok fontos kérdést, amik nagy mértékben segítettek a dokumentum fejlesztését.

13.4. Visszajelzés

Szívesen fogadok minden elektronikus levelet a dokumentum felhasználóitól, a prezentációt érintő fejlesztő javaslatokkal és lehetséges kiegészítésekkel.

13.5. Magyar fordítás

A magyar fordítást Zalányi Balázs Andor (mailto:Zalanyi.Balazs.Andor[kukac]stud.u-szeged[pont]hu) készítette (2004.06.26). A lektorálást Daczi László (mailto:dacas[kukac]fsf[pont]hu) végezte el (2004.06.26). A dokumentum legfrissebb változata megtalálható a Magyar Linux Dokumentációs Projekt (<http://tldp.fsf.hu/>) honlapján. A dokumentum fordítása a Szegedi Tudományegyetem (<http://www.u-szeged.hu/>) nyílt forráskódú szoftverfejlesztés speciálkollégium (<http://www.inf.u-szeged.hu/~havasi/opensource/>)ának segítségével valósult meg.

Megjegyzések

1. Ezen dokumentum erősen támaszkodik a csillaggal jelölt írásokra.